

Категории

- › Уроки HTML (<http://sozdatisite.ru/category/html>)
- › Уроки CSS (<http://sozdatisite.ru/category/css>)
- › Уроки JavaScript (<http://sozdatisite.ru/category/javascript>)
- › Уроки PHP (<http://sozdatisite.ru/category/php>)
- › Уроки MySQL (<http://sozdatisite.ru/category/mysql>)
- › Уроки Wordpress (<http://sozdatisite.ru/category/wordpress>)
- › Уроки Magento (<http://sozdatisite.ru/category/magento>)
- › Программы для создания сайтов (<http://sozdatisite.ru/category/programmy-dlya-sozdaniya-sajtov>)
- › Дополнительные статьи (<http://sozdatisite.ru/category/dopolnitelnye-stati>)
- › Скрипты (<http://sozdatisite.ru/category/skripty>)
- › Уроки Laravel (<http://sozdatisite.ru/category/laravel>)

Контакты

E-mail: sergiu1607@gmail.com



(<https://ru.jooble.org>)

[Главная \(%address%\)](#) > [Уроки Laravel \(http://sozdatisite.ru/category/laravel\)](http://sozdatisite.ru/category/laravel) > Валидация в Laravel с примерами

Валидация в Laravel с примерами

Валидация формы в Laravel

```
$valid = Validator::make($request->all, [  
    'name' => 'required'
```

sozdatisite.ru

Валидация в Laravel с примерами

Оглавление:

- Валидация формы в Laravel
- Написание логики валидации
- Остановка после первой неудачной проверки
- Отображение ошибок валидации
- Валидация через Form Request в Laravel
- Настройка сообщений об ошибках
- Создание валидаторов вручную

Сегодняшняя тема – это инструкция по валидации в Laravel с примерами. Laravel фреймворк предоставляет множество различных способов валидации данных Вашего приложения. По умолчанию базовый контроллер Laravel использует трейт `ValidateRequests`, который обеспечивает удобный способ валидации входящего HTTP-запроса с помощью множества правил валидации.

Мы можем использовать валидацию в Laravel по-разному: либо внутри метода контроллера, либо с помощью класса `FormRequest` для валидации входящих запросов. В этом уроке мы рассмотрим все способы валидации в Laravel.

Валидация формы в Laravel

Первый шаг – установите последнюю версию Laravel с помощью следующей команды.

Используемая версия Laravel на момент написания статьи 5.8.

```
1 | composer create-project --prefer-dist laravel/laravel blog
```

Перейдите в папку проекта и откройте его в вашем IDE или редакторе кода.

Теперь создайте базу данных MySQL, а также подключите её к Laravel. Укажите данные для подключения к базе данных в файле `.env`.

```
1 DB_CONNECTION=mysql
2 DB_HOST=127.0.0.1
3 DB_PORT=3306
4 DB_DATABASE=blog
5 DB_USERNAME=root
6 DB_PASSWORD=root
```

Создайте файл `FormController.php`, используя следующую команду.

```
1 | php artisan make:controller FormController
```

Создайте два метода внутри файла `FormController.php`.

```
1 <?php
2
3 // FormController.php
4
5 namespace App\Http\Controllers;
6
7 use Illuminate\Http\Request;
8
9 class FormController extends Controller
10 {
11     public function create()
12     {
13
14     }
15
16     public function store(Request $request)
17     {
18
19     }
20 }
```

Теперь напишите два маршрута в файле `routes/web.php`.

```
1 // web.php
2 Route::get('form', 'FormController@create')->name('form.create');
3 Route::post('form', 'FormController@store')->name('form.store');
```

Создайте модель и файл миграции, используя следующую команду.

```
1 | php artisan make:model Form -m
```

Напишите следующий код в файле миграции `[timestamp]_create_forms_table.php`, который находится в папке `database/migrations`.

```
1 // create_forms_table.php
2
3 public function up()
4 {
5     Schema::create('forms', function (Blueprint $table) {
6         $table->bigIncrements('id');
7         $table->string('item_name');
8         $table->string('sku_no');
9         $table->integer('price');
10        $table->timestamps();
11    });
12 }
```

Теперь, для того чтобы создать таблицу, запустите миграции командой:

```
1 | php artisan migrate
```

Также, чтобы предотвратить исключение массового присвоения, добавьте свойство `$fillable`, внутри модели `Form`.

```

1 <?php
2
3 // Form.php
4
5 namespace App;
6
7 use Illuminate\Database\Eloquent\Model;
8
9 class Form extends Model
10 {
11     protected $fillable = ['item_name', 'sku_no', 'price'];
12 }

```

Внутри папки resources/views, создайте файл layout.blade.php и добавьте в него следующий код.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Laravel Form Validation Example Tutorial</title>
8     <link href="{{ asset('css/app.css') }}" rel="stylesheet" type="text/css" />
9 </head>
10 <body>
11
12     <div class="container">
13         @yield('content')
14     </div>
15
16     <script src="{{ asset('js/app.js') }}" type="text/js"></script>
17
18 </body>
19 </html>

```

Теперь в той же папке resources/views создайте файл с именем create.blade.php и добавьте в него следующий код.

```

1 <!-- create.blade.php -->
2
3 @extends('layout')
4
5 @section('content')
6 <style>
7     .uper {
8         margin-top: 40px;
9     }
10 </style>
11 <div class="card uper">
12     <div class="card-header">
13         Add Item
14     </div>
15     <div class="card-body">
16         @if ($errors->any())
17             <div class="alert alert-danger">
18                 <ul>
19                     @foreach ($errors->all() as $error)
20                         <li>{{ $error }}</li>
21                     @endforeach
22                 </ul>
23             </div><br />
24         @endif
25         <form method="post" action="{{ route('form.store') }}">
26             <div class="form-group">
27                 @csrf
28                 <label for="name">Item Name:</label>
29                 <input type="text" class="form-control" name="item_name"/>
30             </div>
31             <div class="form-group">
32                 <label for="price">SKU Number :</label>
33                 <input type="text" class="form-control" name="sku_no"/>
34             </div>
35             <div class="form-group">
36                 <label for="quantity">Item Price :</label>
37                 <input type="text" class="form-control" name="price"/>
38             </div>
39             <button type="submit" class="btn btn-primary">Create Item</button>
40         </form>
41     </div>
42 </div>
43 @endsection

```

Теперь, внутри контроллера FormController.php напишите функцию create().

```

1 // FormController.php
2
3 public function create()
4 {
5     return view('create');
6 }

```

Запускаем команду:

```
1 php artisan serve
```

И теперь Вы можете получить доступ к форме по этому адресу: <http://localhost:8000/form> .

Add Item

Item Name:

SKU Number :

Item Price :

Create Item

(<https://sun9-52.userapi.com/c857024/v857024545/61c32/Wq1UkYYrPZU.jpg>)

Рис 1. Форма для добавления товара

Написание логики валидации

Теперь мы можем написать логику валидации внутри функции `store()` из контроллера `FormController`.

```
1 // FormController.php
2
3 public function store(Request $request)
4 {
5     $validatedData = $request->validate([
6         'item_name' => 'required|max:255',
7         'sku_no' => 'required|alpha_num',
8         'price' => 'required|numeric',
9     ]);
10    \App\Form::create($validatedData);
11
12    return response()->json('Данные сохранены успешно');
13 }
```

Как видите, мы передали нужные правила валидации в метод `validate()`. Если валидация не пройдена, то ошибка сгенерируется автоматически. Если валидация пройдена, наш контроллер продолжит работу и сохранит данные в базе данных, и мы получим ответ в формате `json`.

Если мы отправим форму с пустыми полями, то мы в ответ получим ошибки.

Add Item

- The item name field is required.
- The sku no field is required.
- The price field is required.

Item Name:

SKU Number :

Item Price :

Create Item

(<https://sun9-48.userapi.com/c857024/v857024545/61c57/I7aQ4nFblAU.jpg>)

Остановка после первой неудачной проверки

Иногда может потребоваться остановить выполнение правил после первой неудачной проверки. Для этого, в правилах валидации, назначьте атрибут `bail`.

```
1 // FormController.php
2
3 $validatedData = $request->validate([
4     'item_name' => 'bail|required|max:255',
5     'sku_no' => 'required|alpha_num',
6     'price' => 'required|numeric',
7 ]);
```

Отображение ошибок валидации

Laravel автоматически перенаправляет пользователя на предыдущую страницу. Кроме того, все ошибки валидации будут автоматически перенесены во флеш-переменные сессии.

Обратите внимание, что нам не нужно специально привязывать сообщения об ошибках к шаблону в нашем маршруте GET. Это потому, что Laravel будет проверять наличие ошибок в данных сессий и автоматически привязывать их к шаблону, если они доступны.

В представлении, ошибки из сессий, доступны через переменную `$errors`.

Проверяем, существуют ли ошибки в сессии, и выводим их на экран. Данный код добавляем внутри файла `create.blade.php`, в том месте где хотим, чтобы отображались ошибки.

```
1 @if ($errors->any())
2     <div class="alert alert-danger">
3         <ul>
4             @foreach ($errors->all() as $error)
5                 <li>{{ $error }}</li>
6             @endforeach
7         </ul>
8     </div><br />
9 @endif
```

Валидация через Form Request в Laravel

В приведенном выше примере мы написали правила валидации внутри контроллера. Но, мы также можем создать отдельный файл для написания правил валидации.

Для более сложных сценариев валидации Вы можете создать так называемые form requests. Form requests – это специальные классы, которые содержат логику валидации. Для создания класса form request, используйте Artisan-команду `make: request`.

```
1 php artisan make:request FieldRequest
```

В папке `app/Http/Requests` будет создан файл с именем `FieldRequest.php`.

Давайте добавим несколько правил валидации в методе `rules()`.

```
1 // FieldRequest.php
2
3 public function rules()
4 {
5     return [
6         'item_name' => 'bail|required|max:255',
7         'sku_no' => 'required|alpha_num',
8         'price' => 'required|numeric',
9     ];
10 }
```

Если Вы планируете использовать логику авторизации в другом месте Вашего приложения, верните true из метода authorize().

```
1 // FieldRequest.php
2
3 public function authorize()
4 {
5     return true;
6 }
```

Итак, теперь Вам не нужно переписывать эти правила внутри метода store(), из контроллера FormController.php.

Вам необходимо импортировать пространство имен FieldRequest в файл FormController.php и передать FieldRequest в качестве инъекции зависимости внутри функции store().

```
1 <?php
2
3 // FormController.php
4
5 namespace App\Http\Controllers;
6
7 use Illuminate\Http\Request;
8 use App\Http\Requests\FieldRequest;
9
10 class FormController extends Controller
11 {
12     public function create()
13     {
14         return view('create');
15     }
16
17     public function store(FieldRequest $request)
18     {
19         $validatedData = $request->validated();
20         \App\Form::create($validatedData);
21
22         return response()->json('Form is successfully validated and data has been saved');
23     }
24 }
```

Если валидация не пройдена, будет сгенерирован ответ на переадресацию, чтобы вернуть пользователя на предыдущую страницу. Сообщения об ошибках будут сохранены в флеш-переменных сессии, таким образом делая их доступными для отображения.

Настройка сообщений об ошибках

Вы можете настроить сообщения об ошибках, используемые form request, с помощью метода messages. Этот метод должен возвращать массив пар атрибутов / правил и соответствующие сообщения об ошибках.

```
1 // FieldRequest.php
2
3 /**
4  * Get the error messages for the defined validation rules.
5  *
6  * @return array
7  */
8 public function messages()
9 {
10     return [
11         'item_name.required' => 'An Item Name is required',
12         'sku_no.required' => 'An SKU NO is required',
13         'price.required' => 'The price is required',
14     ];
15 }
```

Сохраните файл и снова отправьте форму без каких-либо значений, и Вы увидите эти сообщения об ошибках вместо используемых по умолчанию.

- An Item Name is required
- An SKU NO is required
- The price is required

Item Name:

SKU Number :

Item Price :

Create Item

(<https://sun9-53.userapi.com/c857024/v857024545/61cf9/0l3hrotof2Q.jpg>)

Рис 3. Изменен стандартный текст ошибок валидации

Создание валидаторов вручную

Если Вы не хотите использовать метод `validate()`, то Вы можете создать экземпляр валидатора вручную с помощью фасада `Validator`. Метод `make()` создает новый экземпляр валидатора.

```

1 // FormController.php
2
3 use Validator;
4
5 public function store(Request $request)
6 {
7     $validatedData = Validator::make($request->all(), [
8         'item_name' => 'bail|required|max:255',
9         'sku_no' => 'required|alpha_num',
10        'price' => 'required|numeric',
11    ]->validate();
12
13    \App\Form::create($validatedData);
14
15    return response()->json('Form is successfully validated and data has been saved');
16 }
```

Это даст нам тот же результат. Если Вы хотите создать экземпляр валидатора вручную, но в то же время воспользоваться автоматической переадресацией, Вы можете вызвать метод `validate()` в существующем экземпляре валидатора.

Для дополнительной информации о валидации, смотрите официальную документацию к Laravel (<https://laravel.com/docs/5.8>).

На этом все, статья по валидации в Laravel 5.8 с примерами завершена.

Источник: [Laravel Validation Example | Laravel Form Validation \(https://appdividend.com/2019/03/09/laravel-5-8-form-validation-tutorial-with-example/\)](https://appdividend.com/2019/03/09/laravel-5-8-form-validation-tutorial-with-example/)

Понравилась статья?

Тогда поделитесь ею с друзьями и подпишитесь на новые интересные статьи.

Поделиться с друзьями:

Подписаться на новые статьи:

Введите Ваш E-mail:

Подписаться

Delivered by FeedBurner (<https://feedburner.google.com>)

Поддержите пожалуйста мой проект!

СУММА	СПОСОБ ОПЛАТЫ
<input type="text" value="3"/> \$	<input type="text"/> <input type="text"/> <input type="text"/>
<input type="text" value="Комментарий получателю"/>	ПОЖЕРТВОВАТЬ
<input type="checkbox"/> Я согласен с договором-офертой	




%next_previous_article%

Если у Вас **есть какие-то вопросы или предложения**, то можете писать их в комментариях или мне на почту sergiu1607@gmail.com. И если Вы **заметили какую-то ошибку в статье**, то прошу Вас, **сообщите мне об этом**, и в ближайшее время я всё исправлю.



Добавляйтесь ко мне в друзья в:

-  - ВКонтакте (<https://vk.com/sergiu920>)
-  - Facebook (<https://www.facebook.com/serj.yo.3>)
-  - Одноклассниках (<https://ok.ru/profile/556210623744>)

Добавляйтесь в мои группы:

-  - Группа в ВКонтакте (<https://vk.com/sozdatisite>)
-  - Группа в Facebook (<https://www.facebook.com/groups/1344199978956118/>)
-  - Группа в Одноклассниках (<https://ok.ru/group/55014111641600>)

Подпишитесь на мои каналы:

-  - Мой канал на Youtube (<https://www.youtube.com/channel/UCfy0QfyсpEg99AVsmTELN4g>)
-  - Мой канал на Google+ (<https://plus.google.com/u/0/113882707433828981293>)

Автор статьи: Мунтян Сергей

Копирование материалов с сайта sozdatisite.ru ([//sozdatisite.ru/](https://sozdatisite.ru/)) ЗАПРЕЩЕНО!!!

Дата добавления: %date%

Восстановление
пароля

5 лет назад · 5 комментариев

Прочитав эту статью, Вы узнаете, как реализовать функционал ...

Подтверждение email

6 лет назад · 13 комментариев

В этой статье Вы узнаете, как реализовать функционал ...

Как реализовать
функционал ...

5 лет назад · 1 комментарий

В этой статье Вы узнаете, как правильно реализовать ...

JavaScript пер
...

6 лет назад · 2 комментария

В этой статье Вы узнаете, как объявить переменные и какие типы ...



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS ?

Имя

Share

Лучшие Новые Старые



Dmitriy

🕒 год назад

не работает (php 8.1)

3 0 • Ответить • Поделиться ›



Timofey Panchenko

🕒 год назад

Не работает регистрация, следовательно и авторизация, из-за отказа в доступе к БД пользователю. Код ошибки 1045

1 0 • Ответить • Поделиться ›



Виталий Сапрыгин

🕒 2 месяца назад edited

Если в инспекторе так:

<div class="page-title-head"><h1>Архив категорий: Название рубрики</h1></div>, то можно стилями .page-title-head h1{ visibility: hidden;} .page-title-head h1 span{visibility: visible;}

0 0 • Ответить • Поделиться ›



IcEWaRRiOr

🕒 год назад

Ещё отступ убирает свойство float со значением left заданное для изображения

0 0 • Ответить • Поделиться ›



Подписаться

Privacy

Не продавайте мои данные

Свежие записи

Массивы в PHP (/php/massivy-v-php.html)

Как загрузить файлы на сервер (/php/kak-zagruzit-fajly-na-server.html)

Что такое фронтенд разработка ? (/dopolnitelnye-stati/frontend-razrabotka-sut-i-cena-raboty.html)

Таймер обратного отсчета для сайта (/skripty/kak-sdelat-tajmer-obratnogo-otscheta-na-javascript.html)

Пространства имен в PHP (/php/prostranstva-imen-v-php.html)

Реклама

Самые читаемые статьи

Создание формы регистрации и авторизации (/php/sozдание-formy-registratsii-i-avtorizatsii-v-php.html)

Просмотров: 131507

Подтверждение адреса электронной почты (/php/podtverzhdenie-email.html)

Просмотров: 33768

Создание базы данных mysql в phpmyadmin (/mysql/sozдание-bazy-dannyh-mysql-v-phpmyadmin.html)

Просмотров: 32877

Как решить ошибку "php не является внутренней или внешней командой" ? (/php/reshenie-oshibki-php-ne-yavlyaetsya-vnutrennej-ili-vneshnej-komandoj.html)

Просмотров: 26520

Форма восстановления пароля (/php/vosstanovlenie-parolya.html)

Просмотров: 18477

Подписка на новые статьи

Введите Ваш E-mail:

Подписаться

Delivered by FeedBurner (<https://feedburner.google.com>)



(<http://www.feedvalidator.org/check.cgi?url=//sozdatisite.ru/latest-articles.xml>)

Вы можете **подписаться** на публикации в этих социальных сетях:



- Моя страница (<https://vk.com/sergiu920>)



- Группа в vkontakte (<https://vk.com/sozdatisite>)



- Моя страница (<https://www.facebook.com/serj.yo.3>)



- Группа в facebook (<https://www.facebook.com/groups/1344199978956118/>)



- Моя страница (<https://ok.ru/profile/556210623744>)



- Группа в одноклассниках (<https://ok.ru/group/55014111641600>)



- Google+ (<https://plus.google.com/u/0/113882707433828981293>)



- Мой канал (<https://www.youtube.com/channel/UCfy0QfyfcpEg99AVsmTELN4g>)

